spmoran: An R package for Moran's eigenvector-based spatial regression analysis

Daisuke Murakami
Center for Global Environmental Research, National Institute for Environmental Studies
16-2 Onogawa, Tsukuba, Ibaraki, Japan
E-mail: murakami_daisuke_t@yahoo.co.jp

Table of Contents

1. Introduction

Eigenvector spatial filtering (ESF; e.g., Griffith, 2003), which is also known as Moran's eigenvector mapping (MEM; e.g., Dray et al., 2006), is a regression approach to estimate and infer regression coefficients in the presence of spatial dependence. Recently, ESF is extended to random effects ESF (RE-ESF; Murakami and Griffith, 2015). RE-ESF increases the estimation accuracy of regression coefficients and their standard errors with shorter computational time. RE-ESF is also extended to spatially varying coefficient (SVC) modeling (Murakami et al., 2017). The package "spmoran" provides R functions for fast estimation of ESF and RE-ESF models with/without SVCs.

This tutorial applies ESF and RE-ESF to a land price analysis of flood hazard. The target area is Ibaraki prefecture, Japan. Explained variables are logged land prices in 2015 (JPY/m^2; sample size: 647; Figure 1). Explanatory variables are as listed in Table 1. All these variables are downloaded from the National Land Numerical Information download service (NLNI; http://nlftp.mlit.go.jp/ksj-e/index.html).

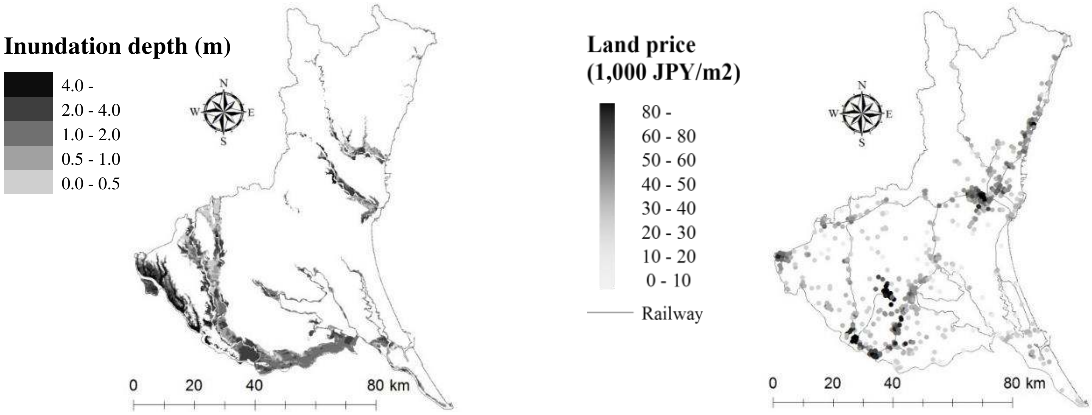

Figure 1. Anticipated inundation depth (left) and officially assessed land prices in 2015 (right) in the Ibaraki prefecture

Table 1. Explanatory variables

Variables	Description
tokyo	Logarithm of the distance from the nearest railway station to Tokyo Station [km]
station	Logarithm of the distance to the nearest railway station [km]
flood	Anticipated inundation depth [m]
city	1 if the site is in an urban promotion land and 0 otherwise

The following is a data image, in which “px” and “py” are spatial coordinates:

```
> data    <- read.csv( "data.csv" )
> data[ 1:6, ]
         px          py     ln_price    station       tokyo   city   flood
1  19235.25   -4784.562   10.126631   4.0109290    43.38504      1     1.5
2  16450.37   -8782.851   10.835652   0.8977986    43.38504      1     0.0
3  17673.30   -8351.802   10.633449   0.5596742    43.38504      1     0.0
4  17824.50   -7704.343    9.878170   0.8504618    43.38504      0     0.0
5  67334.31   58001.724   10.122623   3.1660661   140.95839      1     0.0
6  68929.42   55028.751    9.952278   2.5008292   140.95839      1     1.5
```

ESF/RE-ESF are implemented in the following two steps:

- extraction of Moran's eigenvectors (see Section 2);
- parameter estimation of the ESF/RE-ESF model (see Section 3).

Sections 2 and 3 explain the implementation of these two steps.

2. Extraction of Moran's eigenvectors

Consider a doubly-centered spatial connectivity matrix, $\mathbf{MCM}$, where $\mathbf{C}$ is a symmetric spatial proximity matrix whose diagonals are zeros, $\mathbf{M} = \mathbf{I} - \mathbf{11}'/N$ is a centering operator, where $\mathbf{I}$ is an identity matrix, and $\mathbf{1}$ is a vector of ones, and N is the sample size. The eigenvectors, $\mathbf{E} = \{\mathbf{e}_1,\ldots, \mathbf{e}_N\}$, of $\mathbf{MCM}$ furnish all possible distinct map pattern descriptions of latent spatial dependence, with each level being indexed by the Moran coefficient (MC; Griffith, 2003; Tiefelsdorf and Griffith, 2007). Eigenvectors corresponding to large positive eigenvalue describe map patterns with greater positive spatial dependence (i.e. greater positive MC), whereas eigenvectors corresponding to negative eigenvalue describe map patterns with negative spatial dependence. As positive spatial dependence is dominant in most real-world cases, only eigenvectors with positive eigenvalues are considered in many applied studies.

The function meigen extracts eigenvectors corresponding to positive eigenvalue (i.e. $\lambda_l > 0$, where λ_l is the l-th eigenvalue)[1]. The command is as follows:

```
> coords  <- data[ ,c( "px", "py" ) ]
> meig    <- meigen( coords = coords )
```

Calculated eigenvectors and eigenvalues are displayed by commanding meig$sf and meig$ev, respectively. By default, $\mathbf{C}$ is given by the matrix whose (i, j)-th element equals $\exp(-d_{i,j}/r)$, where $d_{i,j}$ is the Euclidean distance between sites i and j, and r is the longest distance in the minimum spanning tree covering the sample sites (Dray et al., 2006; Murakami and Griffith, 2015).

[1] For the distance-based $\mathbf{C}$, it is standard to set the threshold by $\lambda_l > 0$, which attempts to consider all elements describing positive spatial dependence.

The distance-based **C** may be replaced with other types of spatial connectivity matrix. In this case, user must construct the matrix *a priori*. For example, the following command employs the 4-nearest-neighbor-based **C**:

```
> library( spdep )
> col.knn <- knearneigh( coordinates( coords ), k = 4 )
> cmat    <- nb2mat( knn2nb( col.knn ), style = "B" )
> meigB  <- meigen( cmat = cmat )
```

If the spatial connectivity matrix is not symmetric like the 4-nearest neighbor-based **C**, meigen symmetrizes it by taking$\{\mathbf{C} + t(\mathbf{C})\}/2$. In cases with binary connectivity-based **C** (e.g. proximity-based **C**; *k*-nearest-neighbor-based **C**), $\lambda_l / \lambda_1 > 0.25$ is a standard threshold for the eigenvector extraction[2]. The thresholding is implemented by the following command:

```
> meigB  <- meigen( cmat = cmat, threshold = 0.25 )
```

The eigen-decomposition can be very slow for large samples. To accelerate the computation, the function meigen_f approximates the eigenvectors by applying the Nystrom extension, which is a dimension reduction technique (Murakami and Griffith, 2017)[3]. The command is as follows:

```
> meig_f <- meigen_f( coords = coords )
```

Just like meigen, meig_f$sf and meig_f$ev return approximated eigenvectors and eigenvalues, respectively. By default, the first 200 eigenvectors are approximated[4]. While meigen takes 243.79 seconds for the exact eigen-decomposition, meigen_f takes only 0.38 seconds (see Section 4 for further details).

[2] The threshold $\lambda_l / \lambda_1 > 0.25$ attempts to capture roughly 5% of the variance in explained variables attributable to positive spatial dependence (Griffith and Chun, 2014).

[3] This approximation is available only for the distance-based **C**.

[4] Consideration of 200 eigenvectors is recommended because Murakami and Griffith (2017) show that the approximation error in regression coefficients is quite small when 200 (or more) eigenvectors are considered while the error increases in cases with fewer than 200 eigenvectors.

3. Estimation of ESF and RE-ESF models

3.1. ESF model

The linear ESF model is formulated as follows:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{E}\boldsymbol{\gamma} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}),$$

where $\mathbf{E}$ is a matrix whose l-th column is the l-th eigenvector, $\mathbf{e}_l$, and $\boldsymbol{\gamma}$ is a vector of coefficients. This model is identical to the standard linear regression model.

The ESF model is estimated using the following steps: (i) eigenvectors whose eigenvalue exceeds a threshold are extracted from **MCM**; (ii) stepwise eigenvector selection is performed; (iii) the ESF model with selected eigenvectors is estimated by ordinary least squares.

The following command estimates the linear ESF model. In step (i), following many ESF studies (Griffith, 2003; Tiefelsdorf and Griffith, 2007), eigenvectors whose eigenvalue fulfills $\lambda_l/\lambda_1 >$ 0.25 are extracted from a binary connectivity-based **C** (4-nearest-neighbor-based **C**; see Section 2):

```
> y       <- data[ ,"ln_price" ]                              # Explained variables
> x       <- data[ ,c( "station", "tokyo", "city", "flood " ) ] # Explanatory variables
> meig    <- meigB                                            #Moran's eigenvectors (knn-based C)
> e_res   <- esf( y = y, x = x, meig = meig, vif = 10, fn = "r2" )
```

To cope with possible multicollinearity, eigenvectors are selected so that the variance inflation factor (VIF), which is an indicator of multicollinearity, does not exceed 10. It is implemented by setting vif = 10, whereas VIF is not considered by default. The eigenvector selection is performed by the adjusted R^2 maximization (fn = "r2"; default). Akaike information criterion (AIC) minimization (fn = "aic") or Bayesian Information criterion (BIC) minimization (fn = "bic"). Alternatively, all eigenvectors are considered without selecting them by setting fn = "all".

When fn = "r2", the coefficient estimates yield:

```
> e_res$b
                 Estimate           SE       t_value       p_value
(Intercept)  9.932080e+00 0.0587240255  169.13146372  0.000000e+00
station     -6.911515e-02 0.0065601988  -10.53552610  5.070594e-24
tokyo       -2.846888e-05 0.0004214075   -0.06755664  9.461599e-01
city         6.738630e-01 0.0360500253   18.69244166  2.121536e-62
flood        2.795299e-02 0.0142681894    1.95911280  5.053884e-02
```

Station (-) and city (+) are statistically significant at the 0.1% level. It is verified that urban areas with good access to a railway station have a higher land price than other areas. We can see that flood is positively significant at the 10% level. This result suggests that influence from flood disaster, which is expected to be negative, is not appropriately reflected to land price.

VIF values are displayed by the following command:

```
> e_res$vif
              VIF
station  1.367917
tokyo    1.225594
city     1.282930
flood    1.208189
sf4      1.167728
sf9      1.017697
sf12     1.142611
sf31     1.084662
sf33     1.032077
sf45     1.035118
sf32     1.095973
sf26     1.012234
sf6      1.059948
sf20     1.016059
```

The following command displays error statistics, including residual standard error (residual_SE), adjusted R^2 (adjR2), log-likelihood (logLik), AIC, BIC, and degrees of freedom (DF):

```
> e_res$e
                  stat
resid_SE     0.3542671
adjR2        0.6987400
logLik    -239.0702859
AIC        510.1405718
BIC        581.6981125
```

While we have discussed ESF with binary connectivity-based **C**, which is popular in regional science, ESF with distance-based **C**, which is popular in ecology, is implemented as follows:

```
> meig    <- meigen( coords=coords )                    #Moran's eigenvectors (distance-based C)
> e_res   <- esf( y=y, x=x, meig=meig, fn = "r2" )
```

The distance-based ESF is often referred to as MEM or a principal coordinate neighborhood matrix (PCNM) (see Legendre and Legendre, 2012).

A major disadvantage of ESF is the computational cost. To cope with this problem, Murakami and Griffith (2017) develops a fact approximation. It is implemented by the following command:

```
> meig_f  <- meigen_f( coords = coords )
> e_res   <- esf( y = y, x = x, meig = meig_f, fn = "all" )
```

Here, all eigenvectors in meig_f are considered without selecting them by setting fn = "all". It is sufficient for thousands or more samples (Murakami and Griffith, 2017).

3.2.RE-ESF model

The RE-ESF model is formulated as follows:

$$\mathbf{y} = \mathbf{X\beta} + \mathbf{E\gamma} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\gamma} \sim N(\mathbf{0}, \sigma_\gamma^2\boldsymbol{\Lambda}(\alpha)), \quad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}).$$

Unlike ESF, $\boldsymbol{\gamma}$ is given by a vector of random coefficients: $\boldsymbol{\gamma} \sim N(\mathbf{0}, \sigma_\gamma^2\boldsymbol{\Lambda}(\alpha))$. $\boldsymbol{\Lambda}(\alpha)$ is a diagonal matrix whose elements are the eigenvalues, which are multiplied by α. σ_γ^2 and α represent the variance and the scale of the spatially dependent component; large α implies global-scale spatial variation, while small α implies local variation. These parameters act as shrinkage parameters controlling variance inflation.

The RE-ESF model is estimated using the following steps: (i) eigenvectors whose eigenvalue exceeds a threshold are extracted from **MCM**; (ii) parameters are estimated by the maximum likelihood (ML) method or the restricted maximum likelihood (REML) method. REML estimation is preferable because it accounts for the degrees of freedom lost by estimating the regression coefficients.

The REML estimation is implemented by the following command:

```
> meig    <- meigen( coords = coords )                  #Moran's eigenvectors (distance-based C)
> r_res   <- resf( y = y, x = x, meig = meig, method = "reml" )
```

ML is implemented by replacing method = "reml" with method = "ml".

Estimated coefficients are displayed as follows:

```
> r_res$b
              Estimate       SE           t_value      p_value
(Intercept)   9.9902998898   0.169833051  58.8242385   0.000000e+00
station      -0.0792859163   0.009598674  -8.2600901   8.881784e-16
tokyo        -0.0003715008   0.001795810  -0.2068709   8.361807e-01
city          0.6857752216   0.036926493  18.5713608   0.000000e+00
flood        -0.0043670379   0.014784271  -0.2953841   7.678025e-01
```

Just like the estimates for ESF, station (-) and city (+) are statistically significant, and tokyo is not. In contrast, unlike ESF, flood is not statistically significant. Because RE-ESF tends to outperform ESF in terms of the estimation accuracy of regression coefficients and their standard errors (Murakami and Griffith, 2015), the results of RE-ESF might be more reliable. Error statistics are extracted by the following command:

```
> r_res$e
                   stat
resid_SE           0.3116825
adjR2(cond)        0.7649824
rlogLik         -262.9627231
AIC              543.9254462
BIC              584.1765628
```

where adjR2(cond) is the adjusted conditional R^2, and rlogLik is the restricted log-likelihood. rlogLik is replaced with loglik, which denotes log-likelihood, if method = "ml". It is important to note that, when REML is used, AIC and BIC are comparable only with models with the same explanatory variables. resf also returns the estimated shrinkage parameters as follows:

```
> r_res$s
                  par
shrink_sf_SE      0.4337118
shrink_sf_alpha   0.2449076
```

where shrink_sf_SE and shrink_sf_alpha are σ_γ and α, respectively. The standard error of the spatially dependent component (shrink_sf_SE = 0.4337118)[5] is greater than the residual standard error (resid_SE = 0.3116825). In other words, substantial spatial dependent variations, which are ignored if the linear regression model is estimated, are captured by $\mathbf{E\gamma}$. shrink_sf_alpha is smaller than one. This implies that coefficients on each eigenvector are shrunk comparatively equally, irrespective of their corresponding eigenvalues. The resulting $\mathbf{E\gamma}$ has local-scale spatial variations relative to $\mathbf{E\gamma}$ with large shrink_sf_alpha.

resf performs the computationally efficient ML/REML estimation of Murakami and Griffith (2017). The command is as follows:

```
> meig_f  <- meigen_f( coords = coords )
> r_res    <- resf( y = y, x = x, meig = meig_f, method = "reml" )
```

3.3.RE-ESF model with spatially varying coefficients (SVCs)

Murakami et al. (2017) suggest that RE-ESF-based SVC modeling outperforms geographically weighted regression (GWR), which is the standard SVC modeling approach, in terms of coefficient estimation accuracy and computational time. spmoran provides a function for the RE-ESF-based SVC modeling.

The RE-ESF model with SVC is formulated as follows:

$$\mathbf{y} = \Sigma_k \mathbf{x}_k \otimes \boldsymbol{\beta}_k + \mathbf{E\gamma} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\beta}_k = \beta_{k,0}\mathbf{1}+\mathbf{E}\boldsymbol{\gamma}_k, \qquad \boldsymbol{\gamma}_k \sim N(\mathbf{0}, \sigma_{\gamma,k}^2\boldsymbol{\Lambda}(\alpha_k)), \qquad \boldsymbol{\varepsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I}),$$

where $\boldsymbol{\beta}_k$ is the vector of SVCs on the kth explanatory variables, $\mathbf{x}_k$. $\boldsymbol{\beta}_k$ consists of the constant component, $\beta_{k,0}\mathbf{1}$, and the spatially varying component, $\mathbf{E}\boldsymbol{\gamma}_k$. The latter is modeled by Moran's eigenvectors, $\mathbf{E}$, and their random coefficients, $\boldsymbol{\gamma}_k \sim N(\mathbf{0}, \sigma_{\gamma,k}^2\boldsymbol{\Lambda}(\alpha_k))$. $\boldsymbol{\Lambda}(\alpha_k)$ is a diagonal matrix whose elements are the eigenvalues, which are multiplied by α_k. $\sigma_{\gamma,k}^2$ denotes the variance of the spatially dependent component, $\mathbf{E}\boldsymbol{\gamma}_k$, whereas α_k denotes the spatial scale of the component; large/small α_k implies global/local-scale spatial variation explained by $\mathbf{E}\boldsymbol{\gamma}_k$. These parameters act as shrinkage parameters controlling variance inflation. An interesting point is that, unlike GWR, the RE-ESF-based approach estimates the spatial scale of each SVC using α_k.

[5] The following relationship holds: $Var[\mathbf{E\gamma}]= \mathbf{E\gamma\gamma}'\mathbf{E}' = \sigma_\gamma^2\mathbf{E\Lambda}(\alpha)\mathbf{E}' = \sigma_\gamma^2\hat{\mathbf{C}}_M^\alpha$, where $\hat{\mathbf{C}}_M^\alpha$ is $\mathbf{MC}^\alpha\mathbf{M}$ approximated by the eigenvectors in $\mathbf{E}$. Hence, σ_γ^2 denotes the variance of the spatially dependent component.

In this tutorial, coefficients on station, city, and flood are allowed to vary across geographical space whereas coefficients on Tokyo are not. The command for the SVC model is as

```
> xv      <- x[ ,c( "station", "city", "flood" ) ]          #x with spatially varying coefficients
> xconst  <- x[ , "tokyo" ]                                 #x with constant coefficients
> meig    <- meigen( coords = coords )                      #Moran's eigenvectors (distance-based C)
> rv_res  <- resf_vc( y = y, x = xv, xconst = xconst, meig = meig, method = "reml" )
```

The constant coefficient estimate for tokyo is returned by the following command:

```
> rv_res$b
          Estimate          SE       t_value     p_value
V1   -0.0009924332   0.001782719  -0.5566962   0.5779578
```

As with the output from the RE-ESF model without SVCs, tokyo is not statistically significant. Considering stability and computational cost, it might be preferable to assume SVCs on at most around four explanatory variables, and constant coefficients on the other explanatory variables (see Section 4).

Estimated SVCs and their *p*-values are displayed by the following command:

```
> rv_res$b_vc[ 1:6, ]
  (Intercept)     station      city         flood
1    9.875385 -0.06311678 0.5690735  0.006637360
2   10.278009 -0.11503321 0.8255947  0.005446833
3   10.173544 -0.10025270 0.7743310  0.006120595
4   10.138267 -0.09395701 0.7445319  0.006262945
5   10.207279 -0.10122246 0.5212322 -0.058020901
6   10.258219 -0.08688370 0.5006614 -0.059386765
```

```
> rv_res$p_vc[ 1:6, ]
  (Intercept)     station         city       flood
1           0 0.288107321 1.324063e-06 0.76500129
2           0 0.006000605 6.344747e-11 0.79813709
3           0 0.012536479 6.735013e-10 0.77200809
4           0 0.019306686 3.431317e-10 0.76597759
5           0 0.058577563 5.667240e-05 0.04008828
```

```
6            0    0.124107522    1.405513e-03      0.15243568
```

They can be summarized as follows:

```
> summary( rv_res$b_vc )
   (Intercept)            station              city               flood
 Min.   : 8.909   Min.   :-0.21020   Min.   :-0.02115   Min.   :-0.066797
 1st Qu.: 9.831   1st Qu.:-0.15448   1st Qu.: 0.57226   1st Qu.:-0.049578
 Median :10.062   Median :-0.12184   Median : 0.68319   Median :-0.013046
 Mean   :10.061   Mean   :-0.11572   Mean   : 0.67039   Mean   :-0.021668
 3rd Qu.:10.242   3rd Qu.:-0.07764   3rd Qu.: 0.81286   3rd Qu.: 0.003591
 Max.   :10.946   Max.   : 0.06522   Max.   : 1.06872   Max.   : 0.009442

> summary( rv_res$p_vc )
   (Intercept)      station              city                flood
 Min.   :0     Min.   :0.000001   Min.   :0.0000000   Min.   :0.003934
 1st Qu.:0     1st Qu.:0.001426   1st Qu.:0.0000001   1st Qu.:0.086623
 Median :0     Median :0.010549   Median :0.0000068   Median :0.585556
 Mean   :0     Mean   :0.123792   Mean   :0.0171369   Mean   :0.495177
 3rd Qu.:0     3rd Qu.:0.175201   3rd Qu.:0.0006345   3rd Qu.:0.853780
 Max.   :0     Max.   :0.945239   Max.   :0.9582583   Max.   :0.995845
```

The result suggests that the spatially varying intercept and SVCs on city are positively significant across the target area. station is negatively significant in many sample sites, and flood is statistically insignificant in most sample sites.

Figure 2 displays the estimated coefficients and their statistical significance. Estimated SVCs on station demonstrate that the distance to a railway station has a significant influence on land price in areas along railways. SVCs on city are positively significant across the target area. SVCs on flood suggest that flood risk is negatively significant around Mito city, which is the prefectural capital. Mito city has a long history as a castle town. The negative sign on flood might be because Mito city has adapted to flood disaster in its long history.

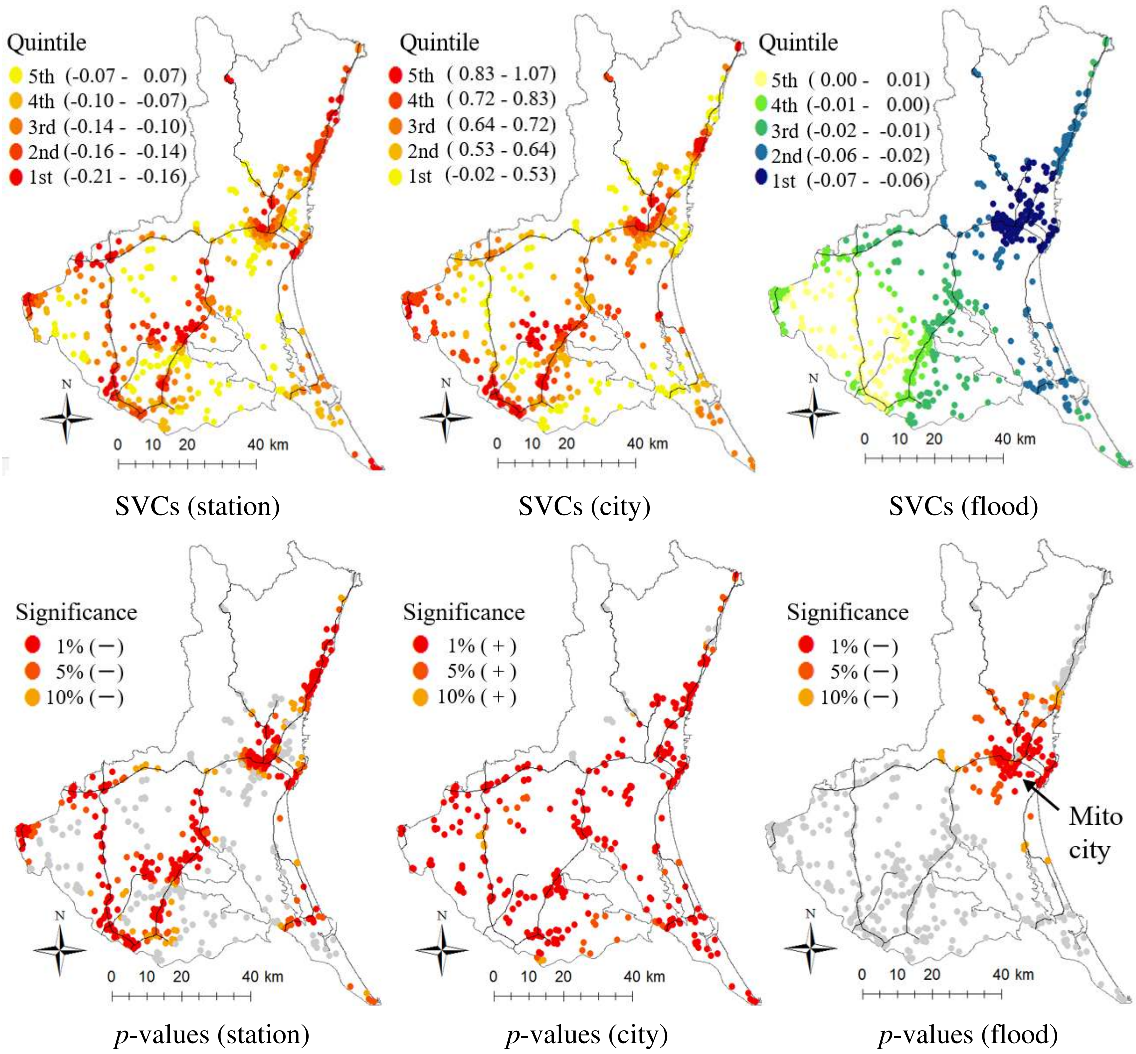

Figure 2. Estimated SVCs and their p-values (the spatially varying intercept is omitted)

Just like resf, resf_vc returns shrinkage parameter estimates for SVCs. In our case, the estimates are as follows:

```
> rv_res$s
                 (Intercept)       station        city       flood
Shrink_sf_SE       0.4562311  0.0820578615  0.30293756  0.04153263
Shrink_sf_alpha    0.1472938  0.0001043149  0.04450748  1.59444026
```

Shrink_sf_SE summarizes the estimated standard errors, $\sigma_{\gamma,k}$, for each SVC, and Shrink_sf_alpha summarizes the estimated α_k parameters. Large α_k values imply strong shrinkage for local variations. For example, SVCs on flood have a global map pattern due to the large α_k value while SVCs on station have a local pattern due to the small α_k value. Thus, the α_k parameter controls the spatial scale of the k-th SVCs.

Error statistics for the SVC model are displayed by the following command although logLik and AIC are reference values because we apply the REML estimation:

```
> r_res$e
                    stat
 resid_SE      0.2637017
 adjR2(cond)   0.8312410
 rlogLik    -230.4469132
 AIC         482.8938264
 BIC         532.0896357
```

4. Tips for fast computation

4.1.Eigen-decomposition

As discussed, meigen_f performs a fact eigen-approximation, and extracts the first 200 eigenvectors by default. The computation is further accelerated by reducing number of approximated eigenvectors. It is achieved by setting enum by a positive integer less than 200. For example, in the case with 5000 samples and enum = 200 (default), 100, and 50, computational times are as follows:

```
> coords_test          <- cbind( rnorm( 5000 ), rnorm( 5000 ) )
```

-----------------CP time (without approximation) -----------------

```
> system.time( meig_test    <- meigen( coords = coords_test ) )
   user   system  elapsed
 242.28     1.44   243.79
```

-----------------CP time (with approximation) ----------------------

```
> system.time( meig_test200 <- meigen_f( coords = coords_test )
   user   system  elapsed
   0.37     0.00     0.38

> system.time( meig_test100 <- meigen_f( coords = coords_test, enum = 100 ) )
   user   system  elapsed
   0.15     0.00     0.16
```

```
> system.time( meig_test50   <- meigen_f( coords = coords_test, enum = 50 ) )
     user      system      elapsed
     0.08        0.00          0.08
```

Figure 3 maps the calculated 1st, 10th, and 100th eigenvectors. It is important to note that, while approximated and exact eigenvectors can have different map patterns respectively, both of them describe patterns in similar spatial scales. In other words, in both cases, 1st eigenvectors describe global map patterns, 10th medium-scale patterns, and 100th local patterns.

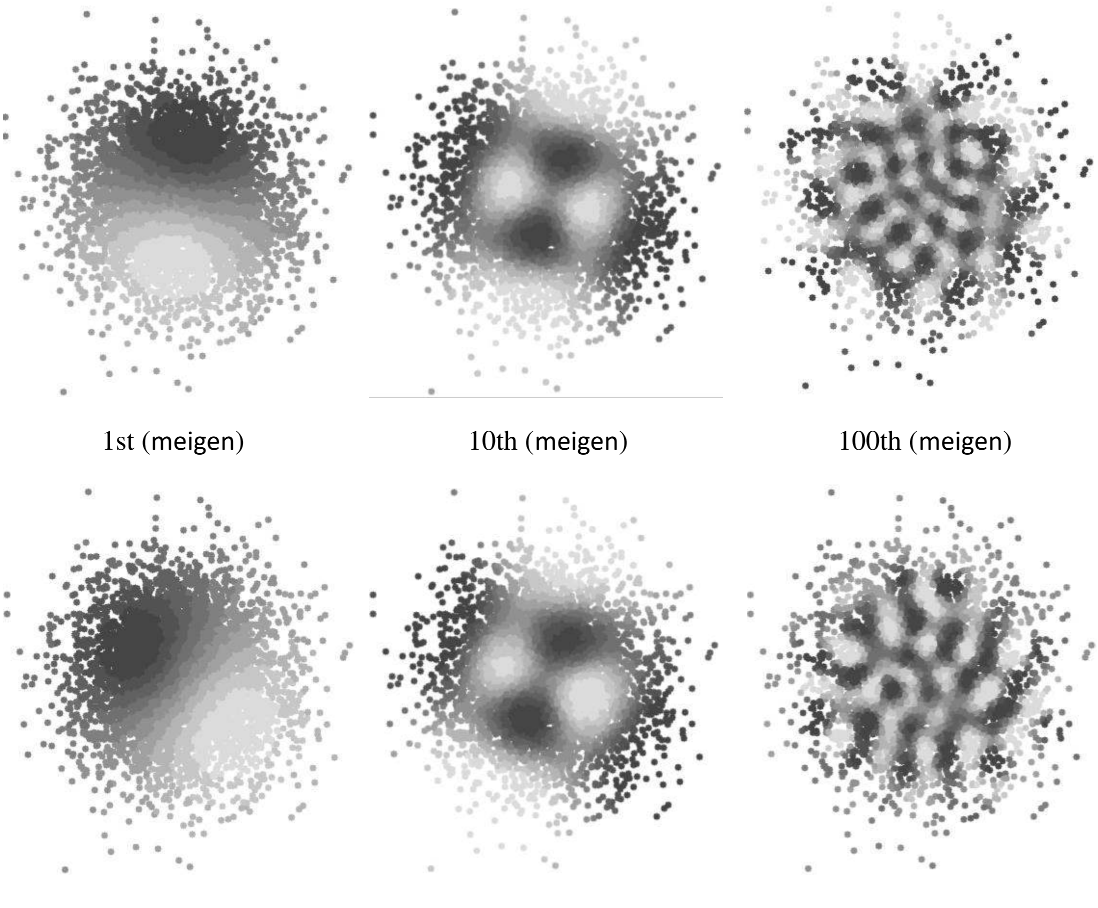

Figure 3. The 1st, 10th, and 100th eigenvectors extracted from meigen and meigen_f

4.2.Parameter estimation

As discussed previously, esf estimates the ESF model is computationally efficiently when fn = "all" (e.g. esf(y = y, x = x, meig = meig_f, fn = "all")). On the other hand, by default, resf estimates the RE-ESF model computationally efficiently.

The RE-ESF-based SVC model is also implemented computationally efficiently with resf_vc. To achieve this, the following conditions must be met: (i) number of eigenvectors considered are reasonably small; (ii) number of SVCs are reasonably small; (iii) standard errors of the SVCs, whose computation is slow for large samples, is not computed. Condition (i) is achieved by setting enum = 200, which is sufficiently small and approximation error is sufficiently small (Murakami and Griffith, 2017). Condition (ii) is achieved by defining x as a matrix, including at most about 4 explanatory variables, and xconst by the other explanatory variables. Condition (iii) is implemented by setting se = F. Finally, the following command implements the SVC model computationally efficiently:

```
-----------------Eigen-decomposition -----------------
> meig    <- meigen( coords = coords, enum = 200)       # slow, but exact
```
or alternatively,
```
> meig    <- meigen_f( coords = coords )                # fast, but approximation

-----------------Parameter estimation -----------------
> xv      <- x[ , c( "x1", "x2", "x3", "x4" ) ]                # at most about 4 explanatory variables
> xconst  <- x[ , c( "x5", "x6", "x7", "x8","x9", "x10") ]  # the other explanatory variables
> rv_res  <- resf_vc( y = y, x = xv, xconst = xconst, meig = meig, method = "reml", se = F )
```

5. Future directions

This tutorial has introduced the R package spmoran. While the functions are still limited, we want to enrich functions relating ESF and RE-ESF gradually.

References

- Dray, S., Legendre, P., and Peres-Neto, P.R. (2006) Spatial modelling: a comprehensive framework for principal coordinate analysis of neighbour matrices (PCNM). *Ecological Modelling*, 196 (3), 483-493.
- Griffith, D.A. (2003) *Spatial autocorrelation and spatial filtering: gaining understanding through theory and scientific visualization*. Springer Science & Business Media.
- Griffith, D., & Chun, Y. (2014). Spatial autocorrelation and spatial filtering. In: *Handbook of regional science*, pp. 1477-1507. Springer.
- Legendre, P. and Legendre, L.F. (2012) *Numerical Ecology*. Elsevier.
- Murakami, D. and Griffith, D.A. (2015) Random effects specifications in eigenvector spatial filtering: a simulation study. *Journal of Geographical Systems*, 17 (4), 311-331.

- Murakami, D. and Griffith, D.A. (2017) Moran's eigenvector-based spatial filtering for large data sets. *Arxiv*, 1702.06220.
- Murakami, D., Yoshida, T., Seya, H., Griffith, D.A., and Yamagata, Y. (2017) A Moran coefficient-based mixed effects approach to investigate spatially varying relationships. *Spatial Statistics*, 19, 68-89.
- Tiefelsdorf, M., and Griffith, D. A. (2007). Semiparametric filtering of spatial autocorrelation: the eigenvector approach. *Environment and Planning A*, 39 (5), 1193-1221.